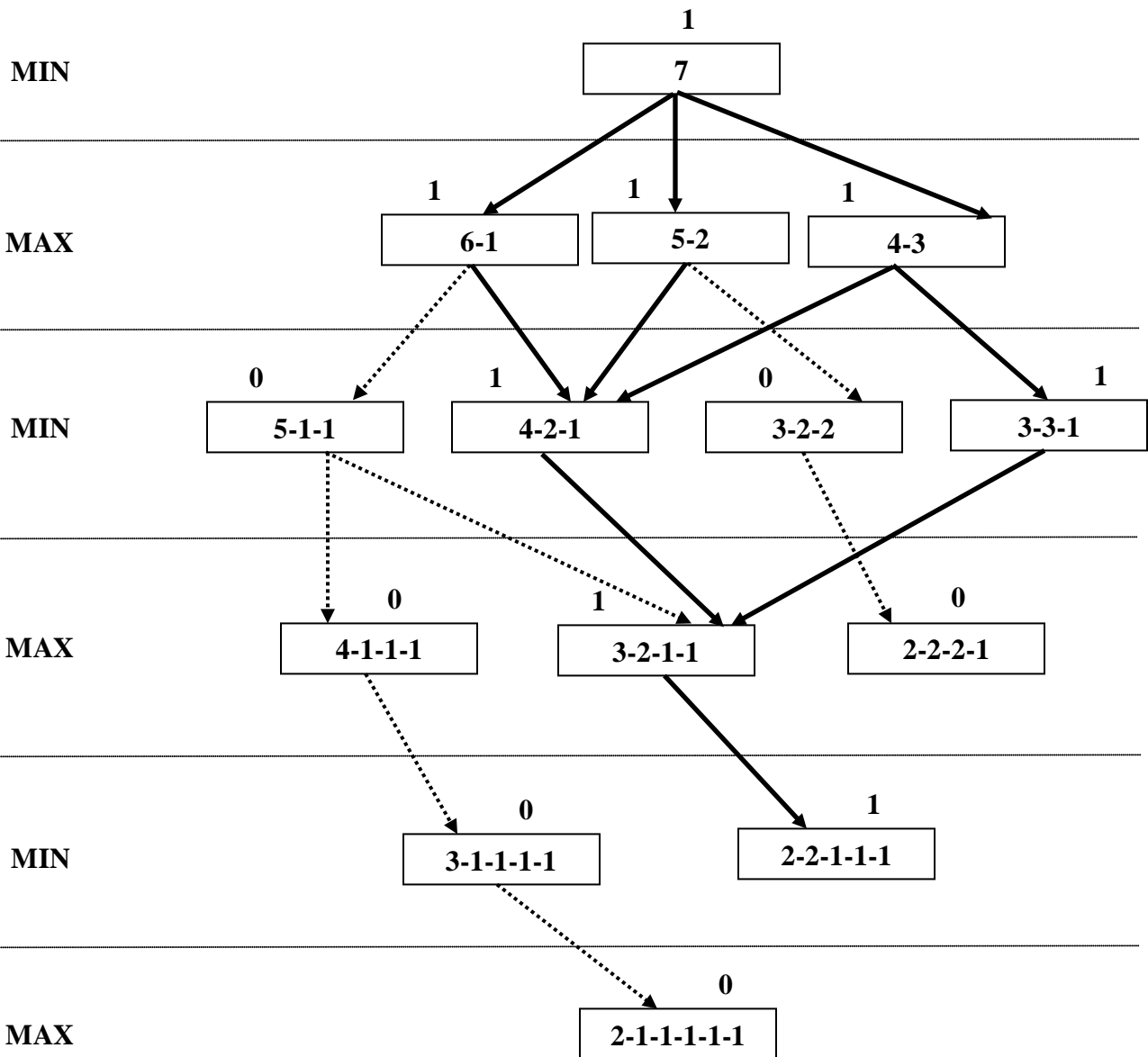


Question 2 – Model Answer

(a) Draw the complete search tree for nim.

The search tree is shown below. The student might draw a search tree which duplicates some of the nodes. This is acceptable.

Note : At this stage, only the search tree is required. The “Min”, “Max” and the Utility Functions are not required.



AI Sheet 9

(b) Assume two players, min and max, play nim (as described above). Min plays first.

If a terminal state in the search tree developed above is a win for min, a utility function of zero is assigned to that state. A utility function of 1 is assigned to a state if max wins the game.

Apply the minimax algorithm to the search tree to assign utility functions to all states in the search tree.

The utility functions are shown in the search tree above. The student should initially assign values to the terminal states and then propagate these up the tree, depending on whether it is min or max's turn to move (min minimises the utility function of its child nodes, max maximises).

The student should mark the tree with the players turn – deduct one mark if this is not done. The other two marks are for correctly assigning the correct utility values to each node. Give one mark if the student appears to be doing it correctly but has made a mistake.

(c) If both min and max play a perfect game, who will win? Explain your answer.

As the value at the top of the search tree is 1, it shows that max will win (if it plays a perfect game). This is because min will never be able to follow a path that leads to a utility function of 0. One mark for pointing this out.

Two marks for showing the path that would be taken, if both players played a perfect game. These are shown in bold in the search tree above.

(d) Given the following search tree, apply the alpha-beta pruning algorithm to it and show the search tree that would be built by this algorithm. Make sure that you show where the alpha and beta cuts are applied and which parts of the search tree are pruned as a result.

The search tree that the student should re-produce is shown below. This example was shown in the lectures and set as an exercise.

If the student produces this search tree they should receive 6 marks (note any extra nodes they produce should be penalised as the idea behind the alpha-beta algorithm is that it restricts the number of nodes that are expanded).

The other three marks are to be allocated based on the way the student explains how the search tree was built. One mark for stating that alpha-beta pruning *must* use Depth First Search.

One mark for explaining why the alpha cut off can be made. One mark why the beta cut off can be made.

AI Sheet 9

This is the explanation given in the course notes

Assume we have evaluated, using depth first search down to node I. This means we can maximise node D to a value of 6. If we now continue the search we will eventually evaluate node J and assign it a value of 8. At this point we do not need to evaluate K (or any of its siblings); for the following reasons.

- Node E already has a value of at least 8 (as MAX is trying to maximise this value so it cannot take on a smaller value).
- Node E is already greater than node D and, as MIN will be trying to minimise these two values, it is always going to choose D over E.
- Therefore, we can cut off the search at this point (shown on the diagram).

Similarly, we can cut off the search from node G downwards.

- Having evaluated F to 2, C cannot be more than this value (as C is trying to minimise).
- B is already greater than 2.
- Therefore, MAX is always going to prefer B over C, so we cannot cut off the rest of the search below C.

